Title:
Canonical Sequence Directed Tactics Analyzer for Computer Go Games

Authors:

1. Chung-Chih Li, Ph.D.
   Assistant Professor
   Computer Science Department
   Lamar University
   Beaumont, Texas 77710, USA

   Tel: 1-409-880-8748
   Fax: 1-409-880-2364
   e-mail: licc@hal.lamar.edu

2. Hikyoo Koh, Ph.D.
   Professor
   Computer Science Department
   Lamar University
   Beaumont, Texas 77710, USA

   Tel: 1-409-880-8779
   Fax: 1-409-880-2364
   e-mail: koh@hal.lamar.edu
Canonical Sequence Directed Tactics Analyzer for Computer Go Games

Chung-Chih Li, Hikyoo Koh

Dept. of Computer Science, Lamar University
Beaumont, Texas, USA

August 24, 2005

Abstract

We present an approach for improving computer Go programs called CSDTA (Canonical Sequence Directed Tactics Analyzer), which analyzes the local tactics on a quadrant of the standard Go board based on a collection of canonical sequences (Joseki). We collect 1278 canonical sequences and their deviations in our system. Instead of trivially matching the current game and the collected canonical sequences, we define a notion of similar sequences with respect to the current game. This paper also explains how to extract the most suitable move from the candidate sequences for the next move. The simplicity of our method and its positive outcome make our approach suitable to be integrated in a complete computer Go program for foreseeable improvement.

Keywords: Computer Go Game

1 Introduction

Go is a two-players board game using identical pieces called stones (each player plays a color, black or white) to occupy territory on the board.1 The game has been very popular in some Asian countries for centuries. In the past few decades, Go has gradually caught much attention in the West. Letting alone the entertaining purpose, to program a computer for playing Go at an acceptable strength has been one of the most challenging tasks in Artificial Intelligence [4]. After Shannon’s first call in [13], computer scientists took “only” fewer than 50 years to have IBM Deep-Blue [6] that defeated the world chess champion, Garry Kasparov, in 1997. In fact, two decades before Kasparov’s sensational defeat, a chess program using straightforward exhaustive search had won the Minnesota Open Chess Championship in 1976 [1]. But, as Berliner predicted in 1978, “even if a full width search program were to become World Chess Champion, such an approach cannot possibly work for Go and this game may have to replace chess as the task par excellence for Artificial Intelligence” [1, 16].

Putting aside the extremely complicate overall strategy of the game (most skillful players believe it’s not computational at all), a simple analysis is enough to convince AI researchers that a classical search-based algorithm doesn’t work for computer Go. The size of the standard Go board is 19 × 19, which is almost six times larger than the chess board. Moreover, the rule is extremely simple. A player can enter a stone at almost everywhere with all kinds of purposes. In other words, the number of available moves in average is about 200 (only about 30 moves available in a typical chess game). As a result, the game-tree with only 4 moves in depth contains about 1.6 billions nodes. Nevertheless, computer Go designers never completely give up the look-ahead technique. For example, the well-known α-β tree is used in a recent investigation [15], where the size of the board is reduced to 5 × 5. The problem is, when the size of the playing board is increased, the “optimized” result in an isolated 5 × 5 territory cannot be extended beyond the boarder; on the contrary, we may just have an opposite result in most nontrivial cases.

Albeit the fact that human brain is not good at mass computation, look-ahead technique is still used by most human Go players; for advanced players, they can check as many as 30 moves ahead [5]. It is clear that such human look-ahead approach must be highly focused, well planned, and goal-directed. With proper conceptual knowledge and goal-directed planning, the game tree

---

1As the West gradually attracted by this fascinating oriental game, there are quite a lot of good introductory books available in English; for example, see [8], or simply type “Go Game” on Amazon’s search engine. A few basic knowledge about the game may be helpful but not essentially needed to appreciate the present paper. We shall skip all details about the game due to the space constrains.
can be subtly pruned to a feasible size. Lehner’s Contingency Plan Goal Tree (CPGT) [10] is a forerunner of such goal-directed planning for computer Go. Another commonly used approach is pattern recognition, where most typical patterns are stored and recognized during the game. This idea can be traced back to Zobrist’s dissertation [18] in 1970. Cazenave’s Gogo [3] is a more recent implementation using the pattern recognition approach. Although computer Go is still in its infancy in terms of the strength, the journey of attempts is rich. There are many other new approaches suggested or implemented for computer Go programs in the past decade, e.g., neural network approach, cognitive model, machine learning, fuzzy logic, and so on. Due to the space constrains, we shall omit them and refer the interested reader to [14] for a partial list of some important computer Go programs between 70’s and late 80’s which are of historical interest and [2, 11] for more recent surveys.

**Our Contribution:** The Canonical Sequence Directed Tactics Analyzer (CSDTA, hereafter) should be considered as a tool to be intergraded in a complete computer Go program. The analysis of CSDTA is based on a rich collection of optimized sequences of moves called canonical sequences (Joseki, Fixed-Stone). While recognizing some predefined patterns to guide the strategic goal or tactical decision during the game is a common approach in computer Go programs, no existent Go program to our knowledge has built a collection of canonical sequences in the program as rich as CSDTA. Also, in order to use the sequences in a more flexible way, we propose an evaluation function to define the similarity between the current game and canonical sequences. The original idea of using canonical sequences was proposed by Koh in [9].

## 2 CSDTA

The underlying idea of CSDTA is that: it recognizes the configuration on the playing board and consults the experiences of expert Go players to make up its decision (next computer move) through some evaluation matrices. The experiences of expert Go players are represented in form of canonical sequences. A canonical sequence is a sequence of local optimal moves with influence that may be extended to remote territory. Each canonical sequence has been thoroughly evaluated by expert players for years, some of them have been considered as classic plays for centuries, and each has proven to be optimal for both players. On the other hand, if one player plays out of the sequence by a mistake or for other trade-off reasons, the opponent will be significantly benefitted if he/she can respond accordingly.

Our database contains 1,278 canonical sequences and their deviations based on the collections in Ishida’s [7] and Ha’s [5]. The deviations are beneficial to the player who play black stones if the opponent plays out of the canonical sequences. Traditionally, canonical sequences are classified into different categories according to their first moves. Each category carries a different agenda for further deployment. We further classify each category in our database according to their second moves as shown in Table 1. The second move of a canonical sequence determines how the opponent engages. Since a canonical sequence can be easily rotated, symmetrically transformed, and switched stone colors, we store only one sequence for all its trivial variations, and black stone always makes the first move.

CSDTA is not a complete Go game program with following restrictions: 1. It only plays the black stone; 2. It considers only one quadrant of the board; 3. It plays only from the deployment stage (opening-game) through the first half of contact-fighting stage (midgame). Moreover, CSDTA does not consider either whole board strategies or the strategies behind each canonical sequence. In a real game, the player determines which canonical sequence to be initiated in a quadrant based on his/her overall deployment strategy. Since CSDTA does not deal with strategy decision, it does not decide which canonical sequence to be used. We simply let the user (or a complete computer Go game program that employs CSDTA) initiate the first move.

No look-ahead algorithm is used in CSDTA, although we believe that including a light look-ahead search could significantly improve the decision quality. Also note that CSDTA is not a miniature of a standard Go game. A quadrant of the standard Go board is not identical to a 10×10 Go board. A 10×10 board is an enclosed world having four corners and four edges, while a quadrant considered by a canonical sequence is a corner and the two open sides opposite to it. More importantly, a corner play usually involves intensive contact fighting with some tricky moves that can’t be found in standard canonical sequences. Another kind of patterns for recognizing urgent moves (Kyuba), vital moves (Kyusho), or tactical moves (Tesuji) are needed for this purpose.

## 3 Evaluation Matrices

The main difference between the evaluation method of CSDTA and most classical pattern recognition approaches used in a computer Go program is that, the sequence to be recognized in CSDTA is a temporal patterns. That is, the developing order of each canonical sequence is essential. CSDTA should shift the focus according to the game being played and try to form
we shall explain how CSDTA enters 9_B in response to 8_W. We have
\[ \Omega = \{(G, 4)_B, (H, 6)_W, (D, 3)_B, (F, 6)_W, (E, 5)_B, (H, 4)_W, (H, 3)_B, (I, 4)_W\} \]
and \( F = (I, 4) \). Here we consider the evaluation of canonical sequences (A) in Figure 2 and have
\[ \omega = \{(G, 4)_B, (H, 6)_W, (D, 3)_B, (H, 3)_W, (H, 4)_B, (I, 4)_W\}. \]
The calculation of the difference degree is rather straightforward: Find the mismatch points between \( \Omega \) and \( \omega \) and sum up the values of \( M_1 \) at the mismatch points with the focus shifted to \( F \). In this example, the mismatch points are \((F, 6)_W, (E, 5)_B, (H, 3)_W, (H, 4)_B\). As the focus shifted to \((I, 4)\), their corresponding values are \(2, 3, 7, \) and \(8\). Thus, the degree of difference is 20. Similarly, one can find the degrees of difference for canonical sequence (B), (C), and (D) in Figure 2, which are \(22, 26\), and \(26\), respectively.

CSDTA computes the difference degree for every sequence with respect to the current game and let \( d \) be the minimum one. All canonical sequences with degree of difference less than or equal to \( d \times (4/3) \) are called *similar canonical sequences* with respect to the current game. Every similar canonical sequences will be passed to stage two for further evaluation. The ratio, \(4/3\), is an experimental value, which is obtained by observing the performance of CSDTA and \(4/3\) provides better results based on our collection. In our example, the canonical sequence (A) in Figure 2 is the one with the minimum degree of difference, which is 20. Therefore, all canonical sequences with degree of difference less than or equal to 26 will be further evaluated in the next stage.

**Stage Two: Selection of the Next Move** Although the canonical sequences selected in stage one

---

<table>
<thead>
<tr>
<th>2nd moves</th>
<th>1st moves</th>
<th>(Total:1278)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-3</td>
<td>-</td>
<td>1</td>
</tr>
<tr>
<td>3-4</td>
<td>-</td>
<td>2</td>
</tr>
<tr>
<td>3-6</td>
<td>-</td>
<td>298</td>
</tr>
<tr>
<td>4-4</td>
<td>10</td>
<td>-</td>
</tr>
<tr>
<td>4-5</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td>4-6</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>5-3</td>
<td>-</td>
<td>267</td>
</tr>
<tr>
<td>5-4</td>
<td>-</td>
<td>182</td>
</tr>
<tr>
<td>6-3</td>
<td>-</td>
<td>25</td>
</tr>
<tr>
<td>6-4</td>
<td>3</td>
<td>36</td>
</tr>
<tr>
<td>Others</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Total</td>
<td>20</td>
<td>514</td>
</tr>
</tbody>
</table>

Table 1: Collection of Canonical Sequences and Deviations
are considered similar, their next moves may be very different. Since the vicinity of a potential move in the current board is critical in determining the goodness of the move, it is natural to put our focus on the move for evaluation, while the last move is still important to us. Therefore, CSDTA uses two focuses in stage two: the last move of the game and the next move of the canonical sequence being evaluated.

Since the next move is much more sensitive to the surrounding than the last move that has been entered, we use a different importance-matrix in which the weight reduced rapidly when the points moving away from the focus (see Figure 1). The pattern is similar to Sander-Davies’s influence pattern [12] and Yen and Hsu’s Rough Inference [17]. Moreover, we think the type of mismatch at each point should be considered. We therefore adjust the weight according to the type of mismatch as shown in Table 2, where \( f(w) \) is defined and \( w \) is the entry of the importance-matrix, \( M_2 \).

The final weight of each selected canonical sequence is simply 1,000 subtracting the sum of adjusted weights in \( M_2 \) as discussed above. Figure 2 shows four similar canonical sequences and their final weights. Consider sequence (A) again. For focus one, the weights corresponding to the four mismatch points are 240, 300, 5, and 5. After adjustment, we have \( 240/2 + 300 \times 2 + 5 + 5 \times 2 = 735 \). For focus two, we have 300, 240, 40, and 30, after adjustment, we have \( 300/2 + 240 \times 2 + 40 + 30 \times 2 = 730 \). Therefore, the final weight of canonical sequence (A) in Figure 2 is 1000 – 735 – 730 = 8535. Similarly, the final weight of canonical sequence (D) is 9406, which turns out to be the one with highest weight among all similar canonical sequences, and hence its next move 11_B at (I,3) becomes CSDTA’s next move 9_B. Note that in (1) we need to find the number of liberties of a cluster. Thus, CSDTA needs to maintain every cluster on the board, which is the only unit higher than a single stone to be recognized by CSDTA. Each cluster is maintained by two linked-lists for connected stones in the same color and their liberties, respectively. It is clear that the cluster is a necessary unit for further developing but not essential in the present paper.

### 4 Assessment

Since CSDTA is not a complete Go game program and no other assistant evaluation is used, it is not appropriate to appraise its strength. For further speculation, however, we shall give it a rough assessment in this section.

#### The Good

Compared to KYU in [12] which is not a complete program either but rated about 30 kyu in the opening game, and it can play about 20 acceptable but, as the authors mentioned, “conservative and unimagination” moves in the whole board (5 or 6 per quadrant), CSDTA plays much better than that. In general, CSDTA can respond with an excellent move if the selected similar canonical sequence has a degree of difference less
than 20, and it can play fairly well when the degree is between 20 and 30. When the degree is between 30 and 40, CSDTA in most cases can still play acceptably. Only when the degree of difference exceeds 40, the quality become unpredictable. In the game shown in Figure 2, the winner, canonical sequence (D), has 26 as the degrees of difference; CSDTA suggests (I,3) as the ninth move of the game. In this example, a good shape for black has successfully deployed.

In most cases, if the minimum degree of difference is less than 30, there always exist canonical sequences with weights higher than 9000. This indicates CSDTA is not likely to suggest a bad move in a normal open game. If the human player does not intentionally play against common canonical sequences, the degrees of difference of the selected similar canonical sequence can be kept under 40 for about 15 moves. In a quadrant, 15 moves would have taken the game into the mid-game stage. Extending to a complete board, CSDTA can play about 60 good moves. We consider to have 60 good moves without too much cost a significant achievement.

The Bad CSDTA mechanically select the “best” canonical sequence without considering the strategy behind the sequence. Also, as no other technique used in CSDTA, it is not surprised that CSDTA may be able to prepare a good tactics deployment for a battle, but it cannot finish the battle and win a territory. Another major shortcoming of CSDTA is that it cannot handle some obvious abnormal moves. On the other hand, we do not really consider these real disadvantages of CSDTA. It should be easy to use the difference degree and weight to set a cutting line for CSDTA to quite and let another method to take on. In fact, many situations in which CSDTA handles badly by their nature should be solved by using another approaches, e.g., search-based algorithms or another kind of patterns as we mentioned earlier at the end of Section 2.

The Cost Memory requirement and response time of CSDTA is not an issue to today’s PC standard. Our program use 150 KB to load all 1278 sequences into main memory. Also, the response time for each move is negligible. Moreover, the memory can be released if the game enters the mid-game stage.

5 Future Study and Conclusion

There are three directions for future development. We briefly discuss them in the following paragraphs.
Overall Game Planning For human player, a canonical sequence is selected according to his/her overall deployment strategy. In other words, each canonical sequence carries a strategy behind. Thus, it is more appropriate if a system can select canonical sequences according to the overall goal. One possible way is to associate a strategy to each canonical sequence in our database. Such strategy can be something like, for example, “this sequence attaches importance to the corner territory” or “this sequence trades the corner territory for influence in the center”. That information will become a built-in strategy for each canonical sequence. Once the system has determined its goal, we should weight the canonical sequences according the goal.

Critical Patterns and Refining Importance-Matrices The knowledge base of CSDTA only contains canonical sequences. If the minimum degree of difference becomes large, CSDTA will still try to extract a similar part from canonical sequence to determine the next move. But, when the minimum degree of difference becomes large, it is more appropriate to consult a different kind of patterns known as "Critical Patterns". In general, such kind of patterns provide urgent moves (Kyuba), vital moves (Kyusho), and tactical moves (Tesuji), which are not a sequence of moves but a few critical moves whenever a certain pattern recognized. Critical Patterns should include information needed to make two eyes, to kill an intervening stones, to connect with friendly groups, etc. Adding some critical patterns should extend the number of acceptable moves significantly.

Also, our importance-matrices are naive in a sense that the degree of importance evenly radiated from a single point (the focus). In human player’s conception, this is not the case. The formation of friend and enemy stones strongly affect the way of the radiation. Thus, to accurate our evaluation, we may adapt the idea of Yen and Hsu’s Rough Inference [17] in designing the importance-matrices according to the shape on the playing board.

Look-ahead Technique Another promising improvement is to adapt a look-ahead program in CSDTA. When the minimum degree of difference is higher than a constant (for example, 40), CSDTA should halt and call that look-ahead program to finish a battle or to actually occupy a territory. In such a way, abnormal moves can also be easily handled.

5.1 Conclusion Since Go game is so sophisticated, it is believed that no single approach can built a satisfactory computer Go program. The goal of CSDTA is to provide a feasible approach in helping improve computer Go programs by using predefined canonical sequences. The notion of similar canonical sequences makes the use of canonical sequences more flexible. We believe that CSDTA has achieved this goal. It is worthwhile to obtain 10 excellent moves for tactical deployment in a quadrant in a cost of 150 KB memory and very insignificant computing time. If CSDTA cooperates with an overall strategic analyzer, stage determining program, focus shifter, and global and local evaluator, it can easily play more than 40 excellent moves on a whole board in an open game.

References